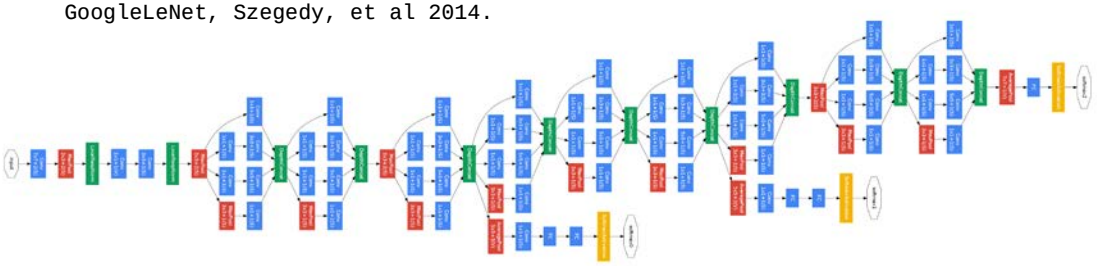
From classical statistics to modern machine learning

Mikhail Belkin
Ohio State University,
Department of Computer Science and Engineering,
Department of Statistics

Simons Institute: Frontiers of Deep Learning
July 2019

GoogleLeNet, Szegedy, et al 2014.



What is new and what is important?

This talk: interpolation and its consequences.

Supervised ML

Input: data $(x_i, y_i),\ i = 1..n\ , x_i \in \mathbb{R}^d, y_i \in \{-1,1\}$

ML algorithm: $f: \mathbb{R}^d \to \mathbb{R}$, that "works" on new data.

Statistical setting:
True/expected risk

Goal of ML: $f^* = \underset{f}{arg\min}\, E_{unseen\ data}\, L(f(x), y)$

(Algorithmic) Empirical risk minimization (ERM) -- basis for nearly all algorithms.

Empirical risk

Goal of ERM: $f^*_{ERM} = \underset{f_w \in \mathcal{H}}{arg\min}\, \frac{1}{n} \sum_{training\ data} L(f_w(x_i), y_i)$

Generalization bounds

WYSIWYG bounds:

VC-dim, fat shattering, Rademacher, covering numbers, margin…

Model or function complexity, e.g., VC or $\|f\|_{\mathcal{H}}$

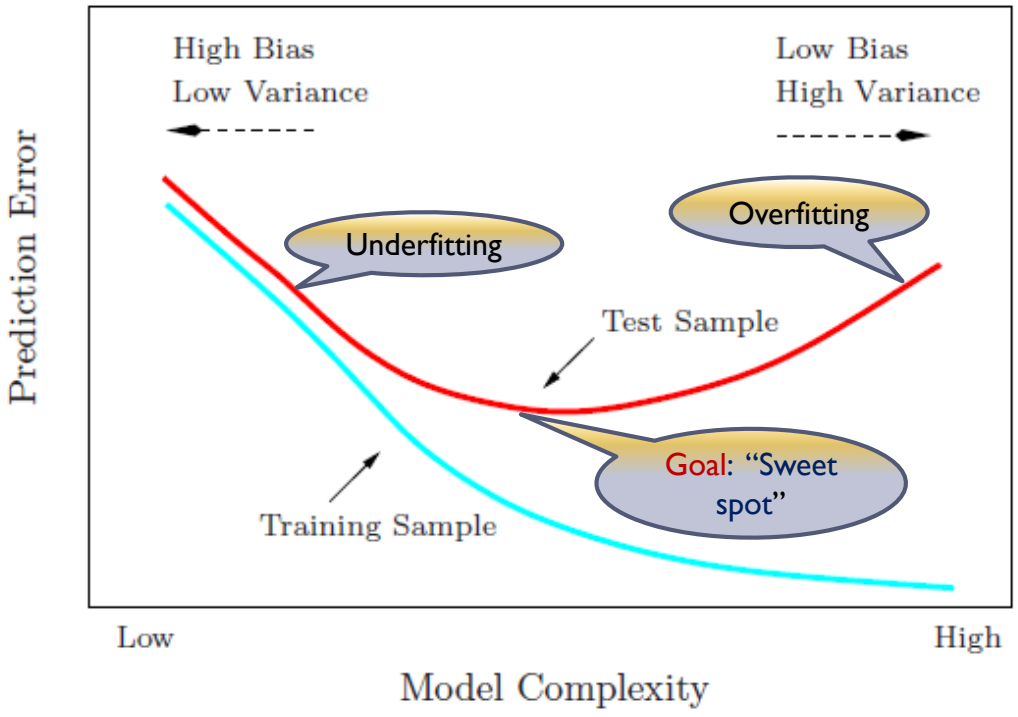
Expected risk: what you get

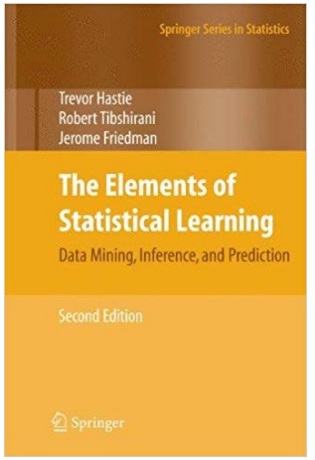
Empirical risk: what you see

$$E(L(f^*, y)) \leq \frac{1}{n}\sum L\left(f^*(x_i), y_i\right) + O^*\left(\sqrt{\frac{c}{n}}\right)$$

Empirical risk approximates expected risk for large n.

Classical U-shaped generalization curve





However, a model with zero training error is overfit to the training data and will typically generalize poorly.

Interpolation

Page 194

Does interpolation overfit?

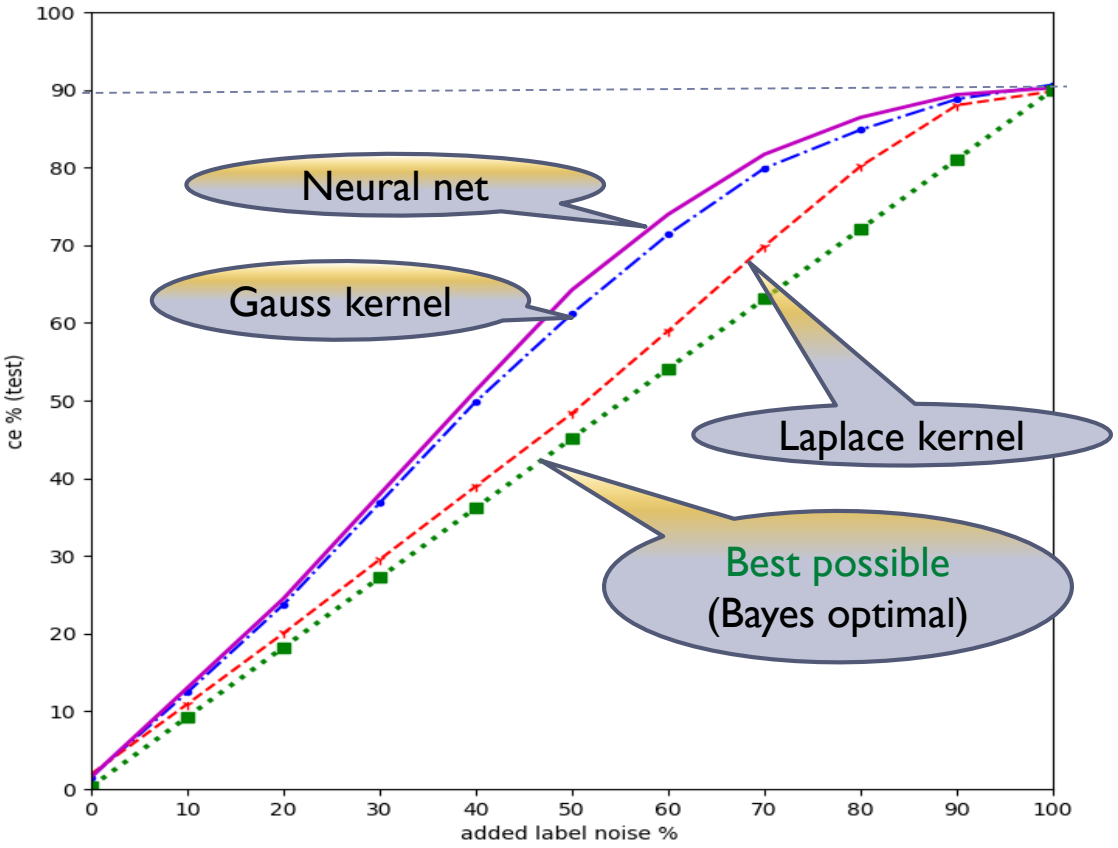
model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75

[CIFAR 10, from *Understanding deep learning requires rethinking generalization*, Zhang, et al, 2017]

But maybe test accuracy should be 100%?
Classical bounds could still apply.

Interpolation does not overfit even for very noisy data

All methods (except Bayes optimal) have zero training *square* loss.



[B., Ma, Mandal, ICML 18]

Deep learning practice

Best practice for deep learning from Ruslan Salakhutdinov's tutorial on deep learning (Simons Institute, Berkeley, 2017):

The best way to solve the problem from practical standpoint is you build a very big system … basically you want to make sure you hit the zero training error.

Leo Breiman
Statistics Department, University of California, Berkeley, CA 94305;
e-mail: leo@stat.berkeley.edu

Reflections After Refereeing Papers for NIPS

Written in 1995

For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

- Why don't heavily parameterized neural networks overfit the data?

Yann Lecun (IPAM talk, 2018):

Deep learning breaks some basic rules of statistics.

It is time to resolve this issue!

This talk

- Statistical theory of interpolation.
  - Why (WYSIWYG) bounds do not apply + what analyses apply.
  - Statistical validity of interpolation.
- The generalization landscape of Machine Learning.
  - Double Descent: reconciling interpolation and the classical U curve.
  - Occams razor: more features is better.
- Interpolation and optimization
  - Easy optimization + fast SGD (+ good generalization).
  - Learning from deep learning for efficient kernel machines.

Classical bounds:

VC-dim, fat shattering, Rademacher, covering numbers, margin…

Model or function complexity, e.g., VC or $\|f\|_{\mathcal{H}}$

Expected risk Empirical risk

$$E(L(f^*, y)) \leq \frac{1}{n}\sum L(f^*(x_i), y_i) + O^*\left(\sqrt{\frac{c}{n}}\right)$$

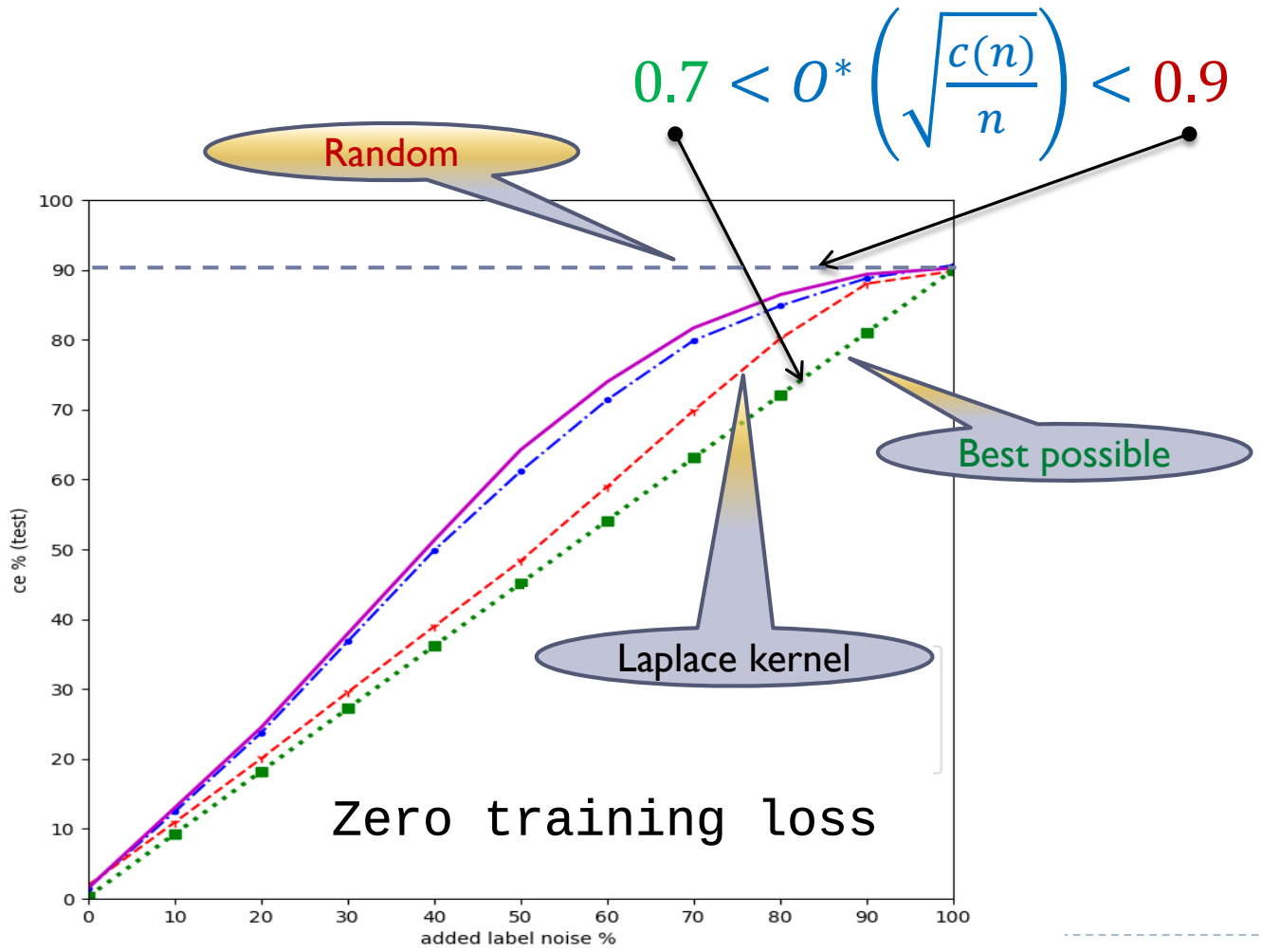
$= 0$

Interpolation

Can such bounds explain generalization?

Bounds?

What kind of generalization bound could work here?



Not a question of improving bounds

correct

$$0.7 < O^*\left(\sqrt{\frac{c(n)}{n}}\right) < 0.9 \qquad n \to \infty$$

nontrivial

There are no bounds like this and no reason they should exist.

A constant factor of 2 invalidates the bound!

Generalization theory for interpolation?

What theoretical analyses do we have?

- VC-dimension/Rademacher complexity/covering/margin bounds.
  - Cannot deal with interpolated classifiers when Bayes risk is non-zero.
  - Generalization gap cannot be bound when empirical risk is zero.
- Regularization-type analyses (Tikhonov, early stopping/SGD, etc.)
  - Diverge as $\lambda \to 0$ for fixed n.
- Algorithmic stability.
  - Does not apply when empirical risk is zero, expected risk nonzero.

WYSIWYG bounds:

training loss $\approx$ expected loss

- Classical smoothing methods (i.e., Nadaraya–Watson).
  - Most classical analyses do not support interpolation.
  - But 1-NN! (Also Hilbert regression Scheme, [Devroye, et al. 98])

Oracle bounds

expected loss $\approx$ optimal loss

A way forward?

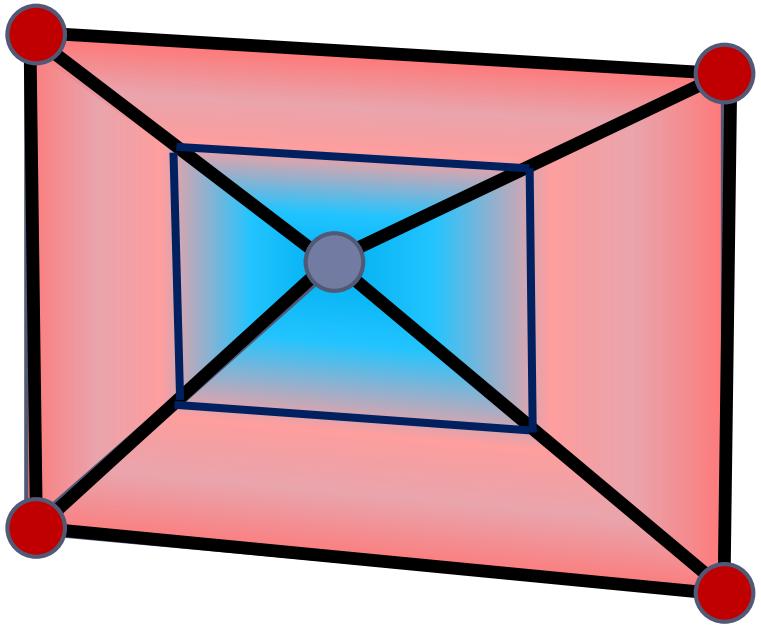
1-nearest neighbor classifier is very suggestive.

Interpolating classifier with a non-trivial (sharp!) performance guarantee.

Twice the Bayes risk [Cover, Hart, 67].

- Analysis not based on complexity bounds.
- Estimating expected loss, not the generalization gap.

Simplicial interpolation



1. Triangulate.
2. Linearly interpolate
3. Threshold

[B., Hsu, Mitra, NeuriPS 18]

Nearly optimality of SI

Theorem: (dimension d) (additional cond. to get exp).

$$E\big(L(SI)\big) - Bayes\ Risk < \boxed{\frac{1}{2^d} \times Bayes\ Risk}$$

Cf. classical bound for 1-NN:

$$E\left(L\left(1_{NN}\right)\right) - Bayes\ Risk < \boxed{Bayes\ Risk}$$

The blessing of dimensionality.

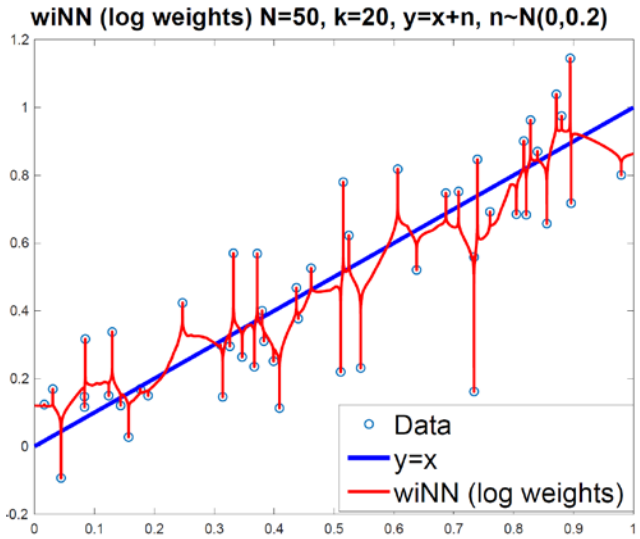
[B., Hsu, Mitra, NeuriPS 18]

Interpolated k-NN schemes

$$f(x) = \frac{\sum y_i k(x_i,x)}{\sum k(x_i,x)}$$

$$k(x_i,x) = \frac{1}{||x-x_i||^\alpha}, \; k(x_i,x) = -\log||x-x_i||$$

(cf. Shepard's interpolation)



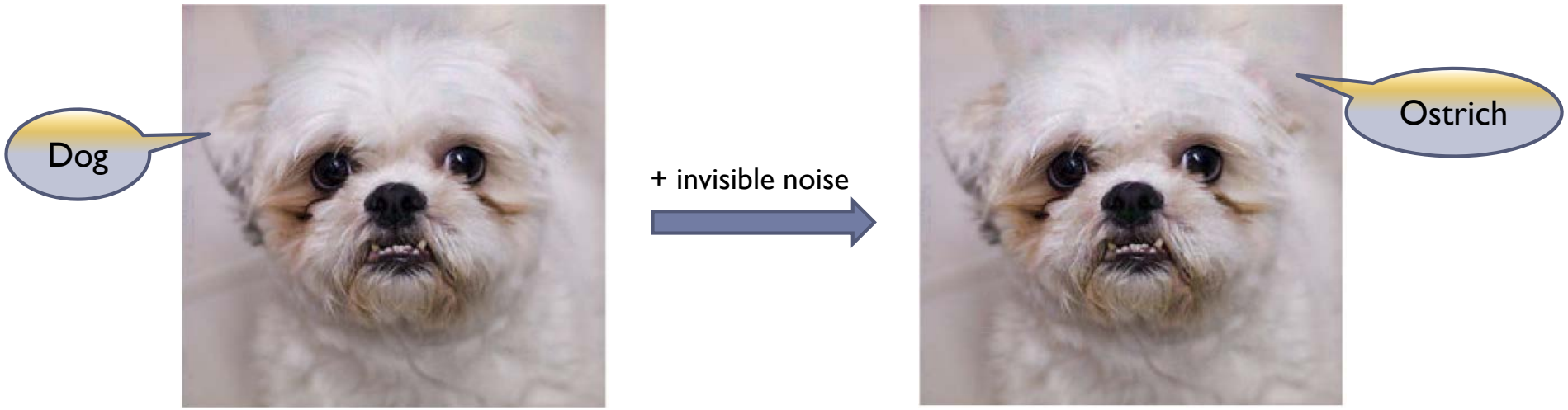
Theorem:

Weighted (interpolated) k-nn schemes with certain singular kernels are consistent (converge to Bayes optimal) for classification in any dimension.

Moreover, statistically (minimax) optimal for regression in any dimension.

[B., Hsu, Mitra, NeuriPS 18] [B., Rakhlin, Tsybakov, AIStats 19]

Interpolation and adversarial examples



From Szegedy, at al, ICLR 2014

Theorem: adversarial examples for interpolated classifiers are asymptotically dense (assuming the labels are not deterministic).

[B., Hsu, Mitra, NeuriPS 18]

This talk so far

A. Empirical validity of interpolation.
   - Neural network and kernel machines.

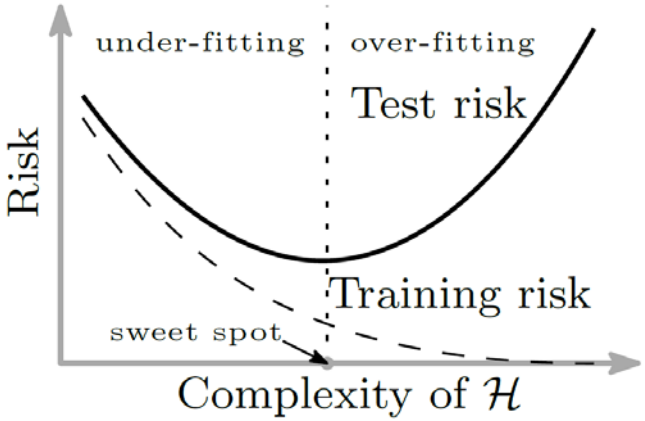
B. Statistical validity of interpolation.
   - Statistically optimal interpolating rules.

But how do A (ERM) an B (non-ERM) relate?

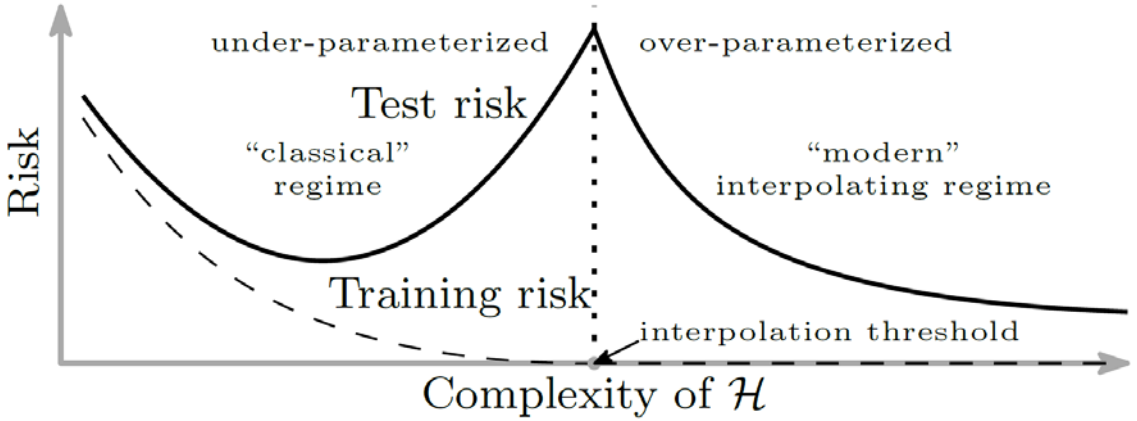
Specifically, how is ERM generalization affected by model complexity?

“Double descent” risk curve

Classical risk curve

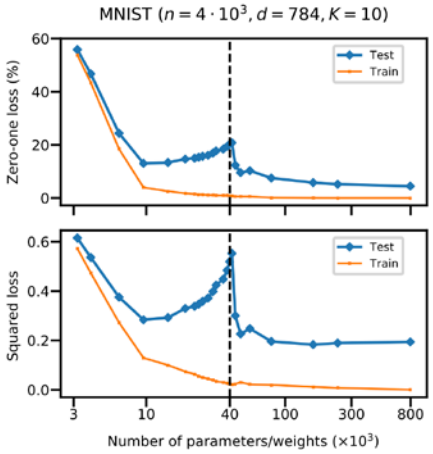


New “double descent” risk curve

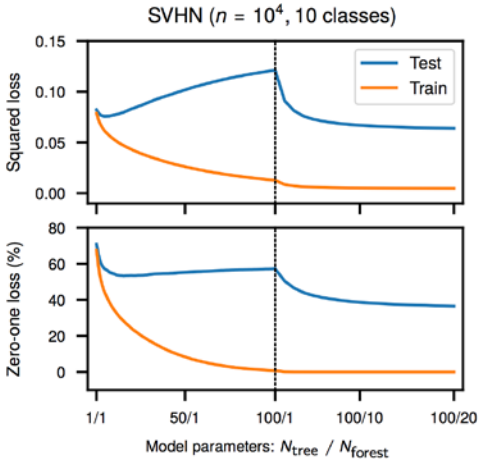


[B., Hsu, Ma, Mandal, PNAS, to appear]

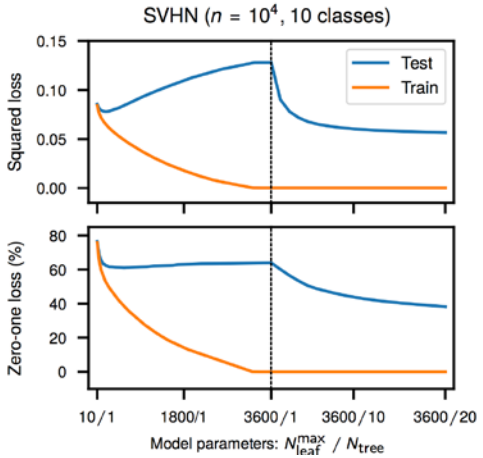
Fully connected network

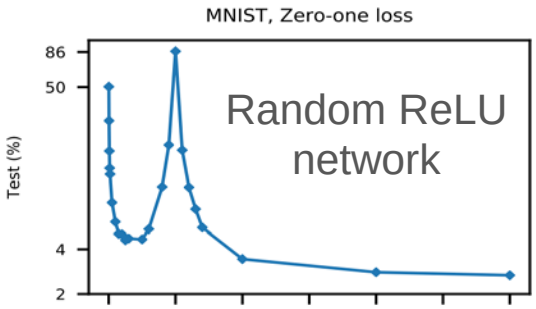


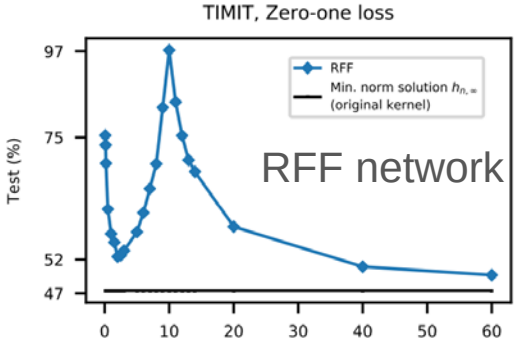
Random Forest

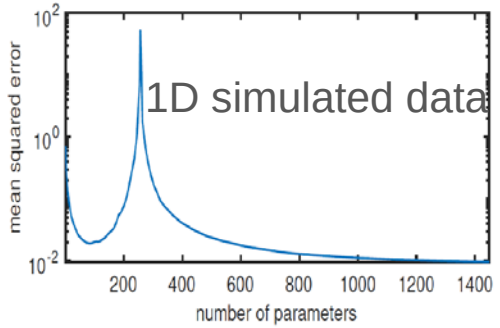


L2-boost

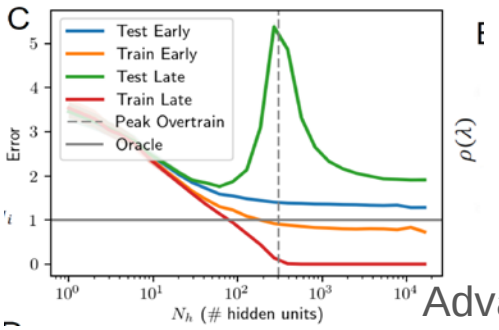




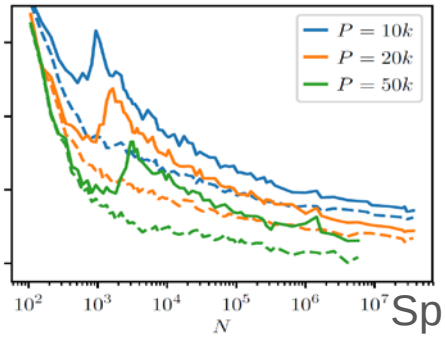




[B., Hsu, Ma, Mandal, 18]

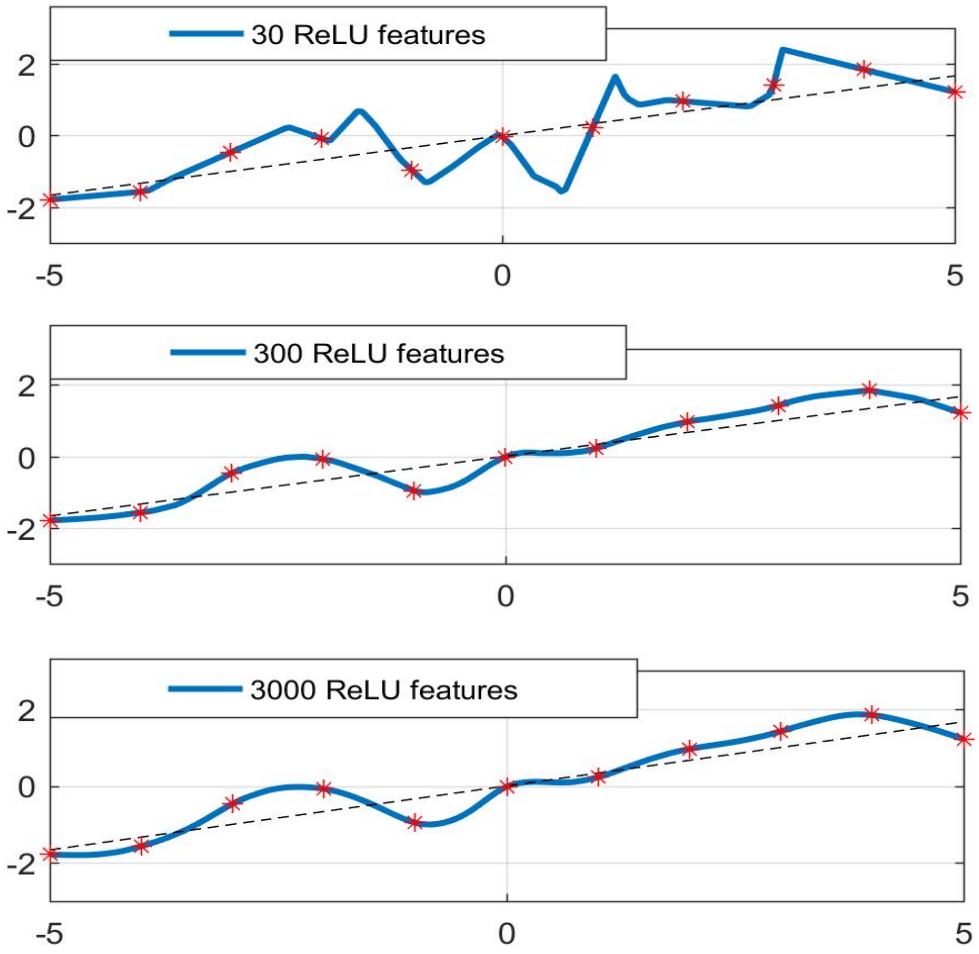


Advani, Saxe, 2017



Spigler, et al, 2018

More parameters are better: an example



RFF networks

Feature map $\phi: \mathbb{R}^d \to \mathbb{R}^N$, w_j sampled iid from normal distribution in $\mathbb{R}^d$.

$$\phi(x) = (e^{i\pi\langle w_1, x\rangle}, \dots, e^{i\pi\langle w_N, x\rangle})$$

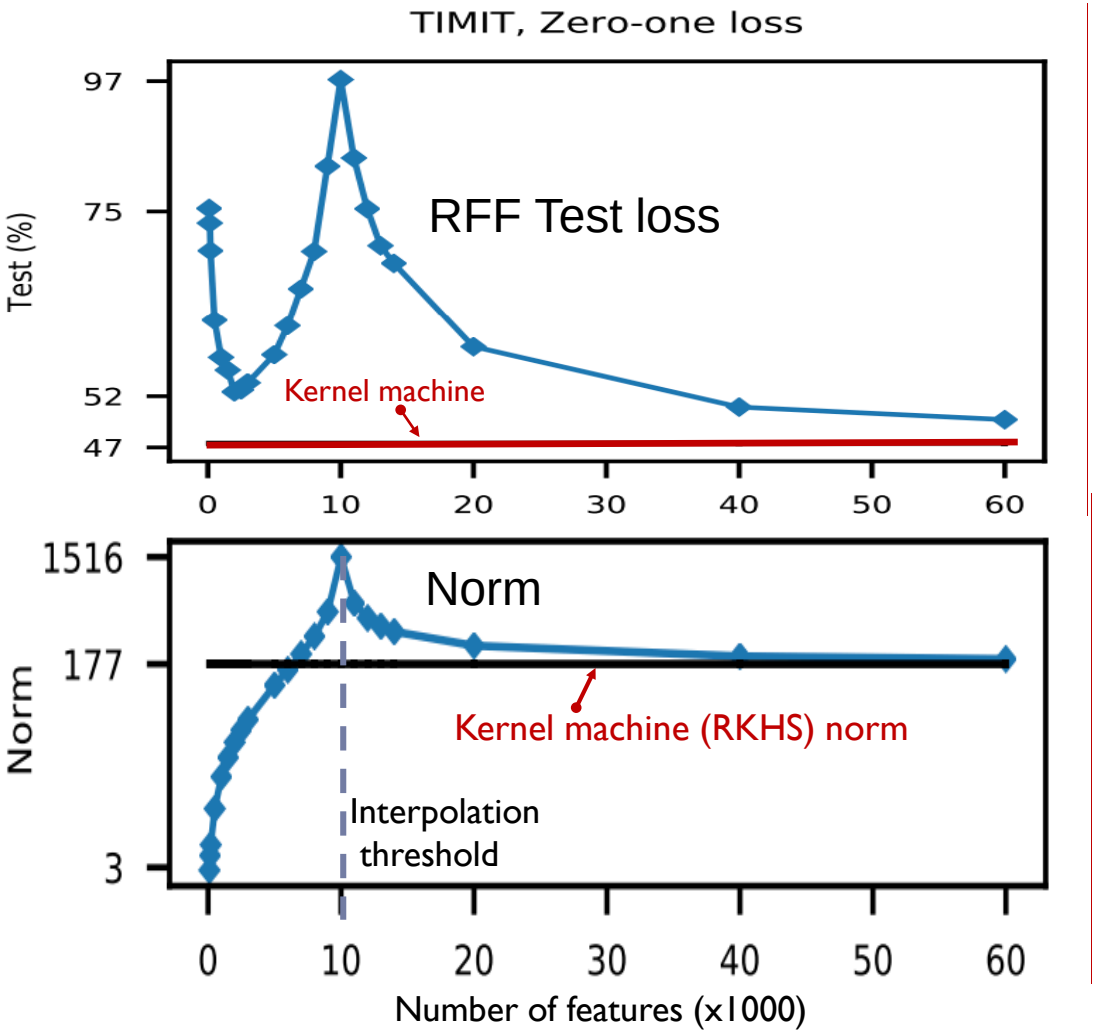
Random Fourier Features (RFF) [Rahimi, Recht, NIPS 2007]

Followed by linear regression.

$$h_{n,N}(x) = \sum_{j=1}^{N} \alpha_j \, e^{i\pi\langle w_j, x\rangle}$$

Neural network with one hidden layer, cos non-linearity, fixed first layer weights. Hidden layer of size N.

What is the mechanism?



$N \to \infty$ -- infinite neural net

=

kernel machine

=

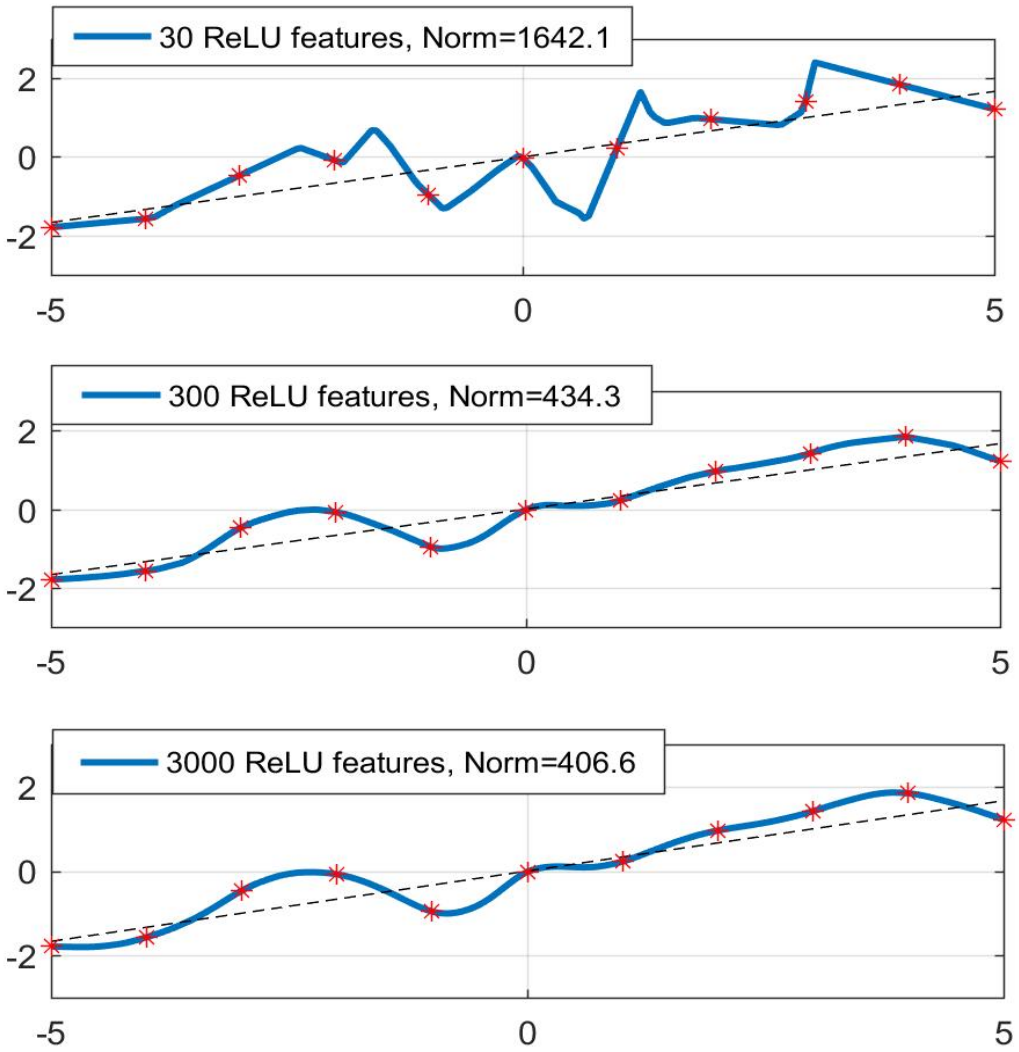
minimum norm solution

$argmin_{h \in \mathcal{H},\, h(x_i)=y_i} \; ||h||_{\mathcal{H}}$

More features ⇒

better approximation
to minimum norm solution

Smaller norm ensures smoothness



Is infinite width optimal?

Infinite net (kernel machine) $h_{n,\infty}$ is near-optimal empirically.

Suppose $\forall_i \;\; y_i = h^*(x_i)$ for some $h^* \in \mathcal{H}$ (Gaussian RKHS).

Theorem (noiseless case):

$$\left|h^*(x) - \; h_{n,\infty}(x)\right| < \; Ae^{-B(n/\log n)^{1/d}} \; ||h^*||_{\mathcal{H}}$$

Compare to $O\left(\frac{1}{\sqrt{n}}\right)$ for classical bias-variance analyses.

[B., Hsu, Ma, Mandal, 18]

Teacher/student

Theorem:

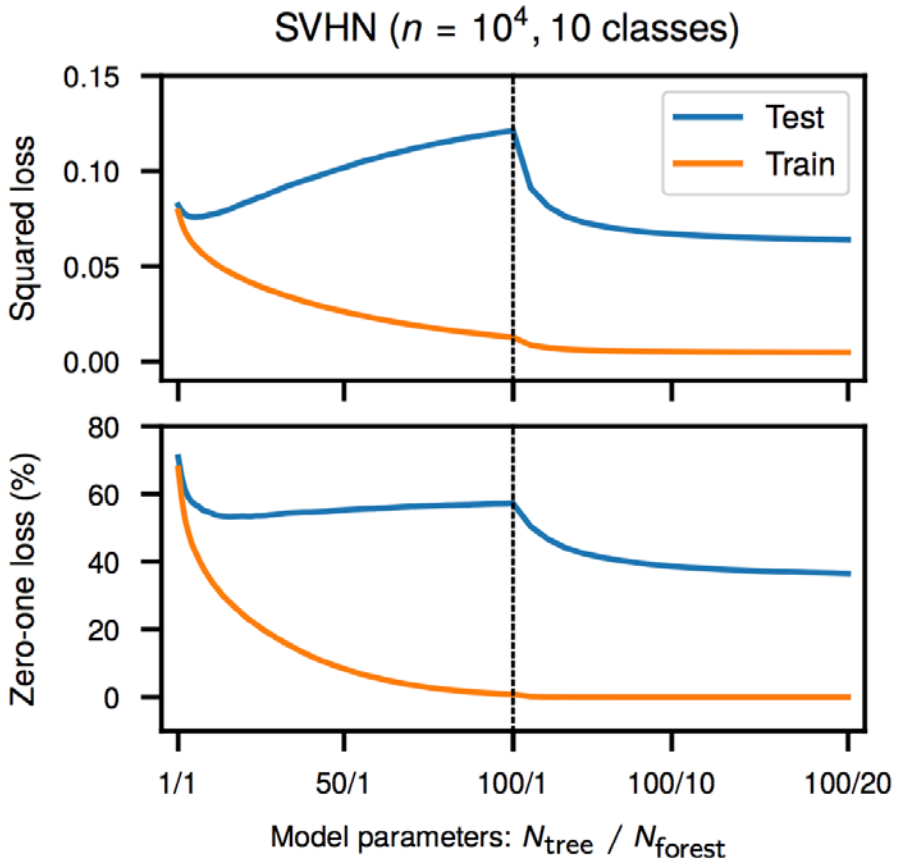
$$\left|h^*(x) - h_{n,\infty}(x)\right| < Ae^{-B(n/\log n)^{1/d}} ||h^*||_{\mathcal{H}}$$

Teacher Student

Student learns from teacher yet the network structure is completely different.

Teacher/student setting is classical approximation theory.

Smoothness by averaging

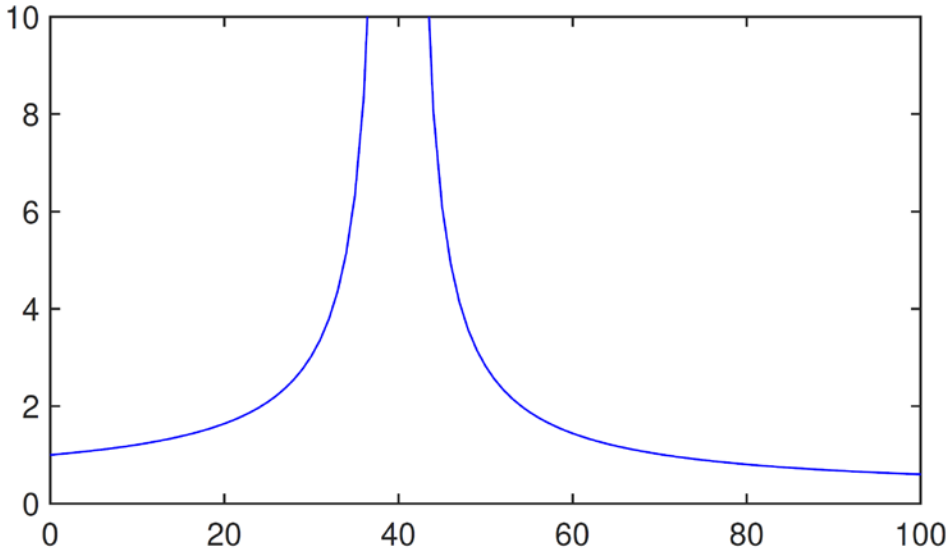


An average of interpolating trees is better than any individual tree.

Cf. PERT [Cutler, Zhao 01]

Double Descent in Linear Regression

Choosing maximum number of features is optimal under the "weak random feature" model.



[B., Hsu, Xu, 19].

Related work:
[Bartlett, Long, Lugosi, Tsigler 19],
[Hastie, Montanari, Rosset, Tibshirani 19] [Mitra, 19],
[Muthukumar, Vodrahalli, Sahai, 19]

Occams's razor

Occam's razor based on inductive bias:

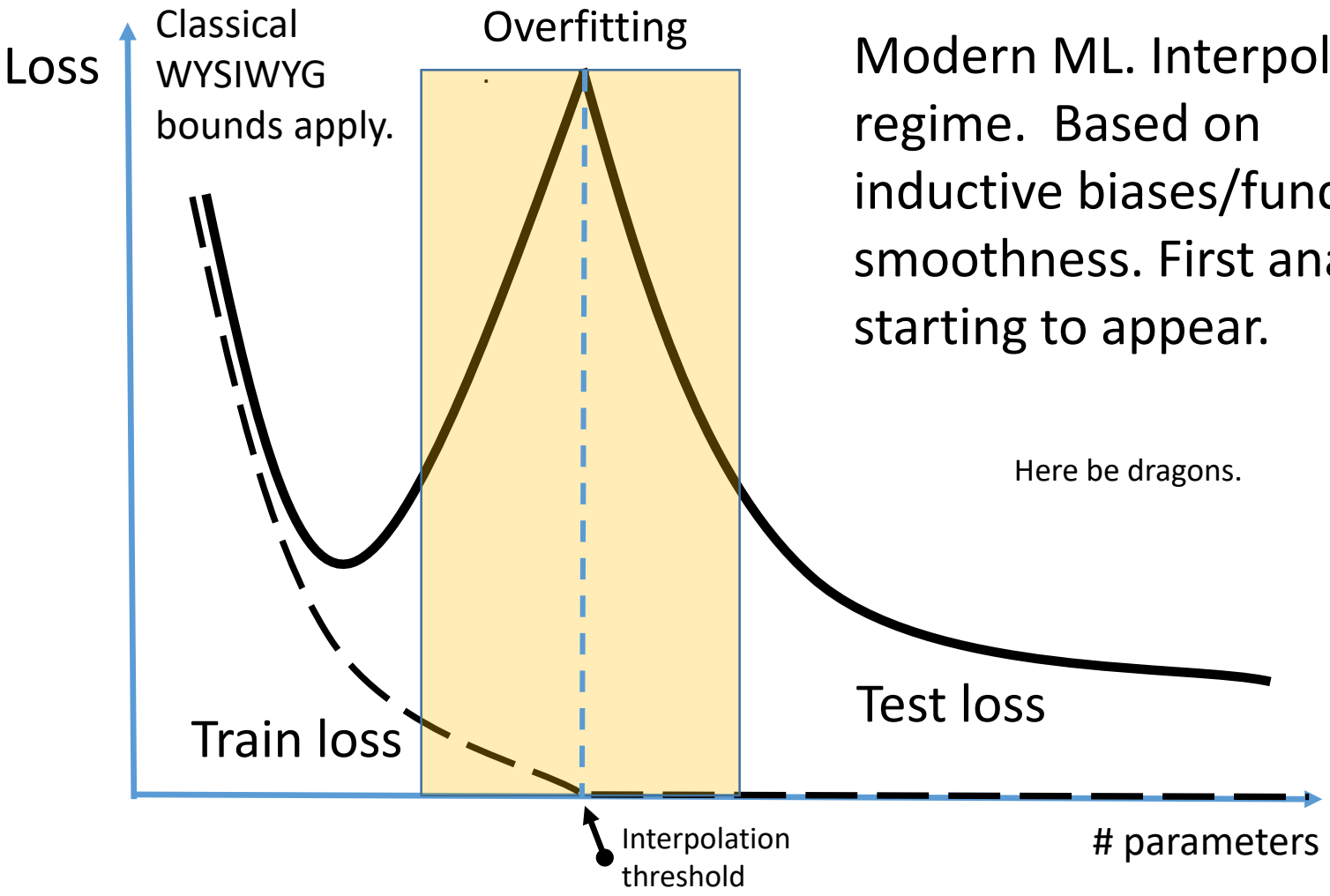
Choose the smoothest function subject to interpolating the data.

Three ways to increase smoothness:

- Explicit: minimum functional norm solutions
  - Exact: kernel machines.
  - Approximate: RFF, ReLU features.
- Implicit: SGD/optimization (Neural networks)
- Averaging (Bagging, L2-boost).

All coincide for kernel machines.

The landscape of generalization



Modern ML. Interpolation regime. Based on inductive biases/functional smoothness. First analyses starting to appear.

Here be dragons.

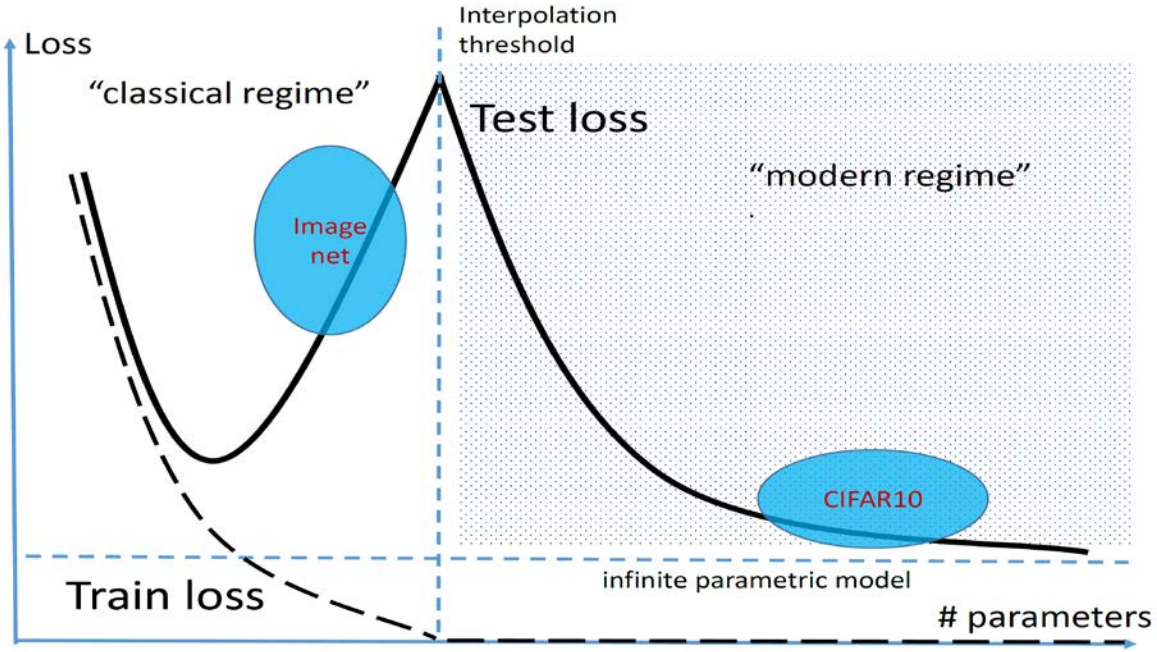
Where is the interpolation threshold?

What is over-parametrization?

Interpolation threshold #parameters = #data * #classes

CIFAR10 #data=50K, #classes=10

Imagenet #data=1M, #classes=1000



This talk

- Statistical theory of interpolation.
  - Why (WYSIWYG) bounds do not apply + what analyses do apply.
  - Statistical validity of interpolation.

- The generalization landscape of Machine Learning.
  - Double Descent: reconciling interpolation and the classical U curve.
  - Occams razor: more features is better.

- Interpolation and optimization
  - Easy optimization + fast SGD (+ good generalization).
  - Learning from deep learning for efficient kernel machines.

Optimization under interpolation

Classical (under-parametrized):

- Many local minima.
- SGD (fixed step size) does not converge.

Modern (interpolation).

- Every local minimum is global.

A lot of recent work. [Kawaguchi, 16] [Soheil, et al, 16] [Bartlett, et al, 17] [Soltanolkotabi, et al, 17, 18] [Du, et al, 19] …

- Small batch SGD (fixed step size) converges as fast as GD.

[Ma, Bassily, **B**., ICML 18]

Why SGD?

$$w^* = \operatorname*{argmin}_w L(w) = \operatorname*{argmin}_w \frac{1}{n} \sum L_i(w)$$

SGD Idea: optimize $\sum L_i(w)$, m at a time.

Error after t steps

GD: e^{-t}

SGD: $1/t$

What is the reason for practical success?

All major neural network optimization use SGD.

SGD is not simply noisy GD.

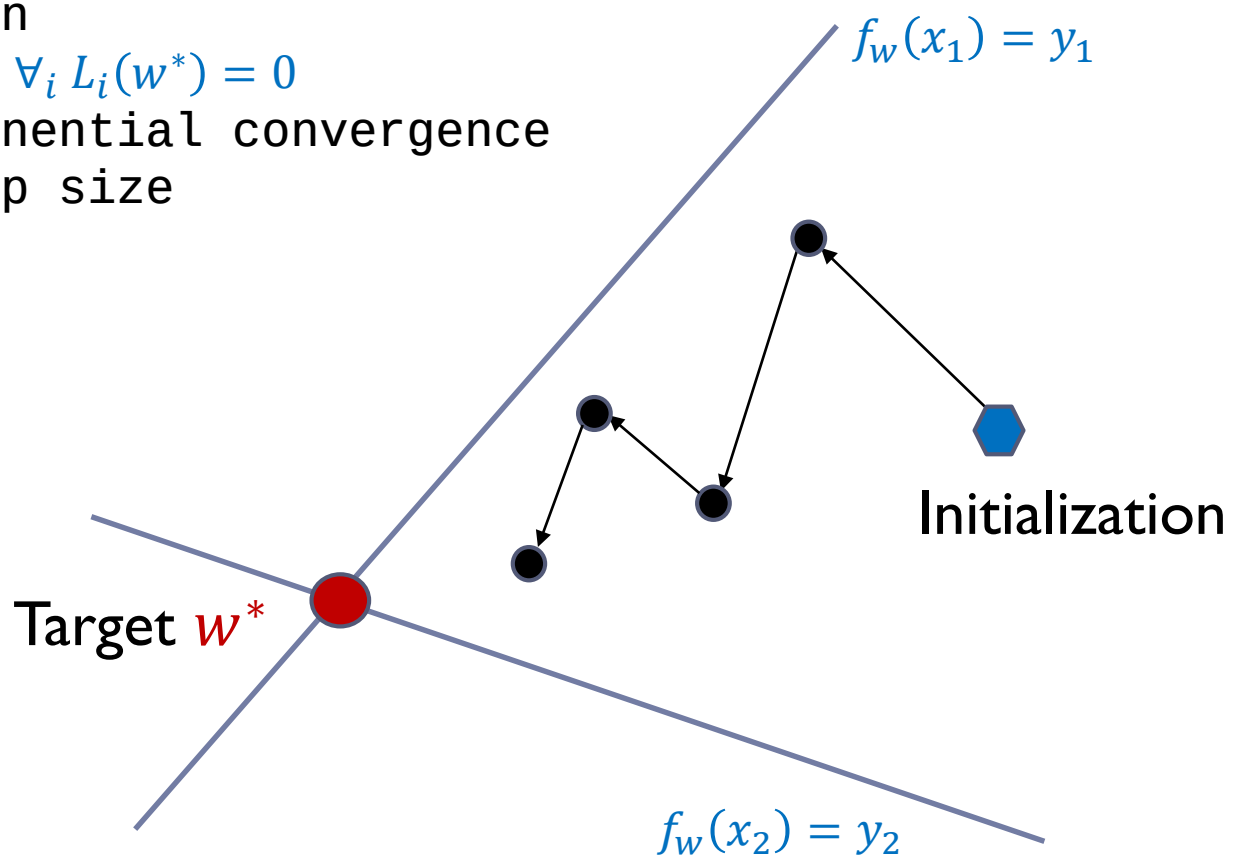
SGD under interpolation

Key observation:

Interpolation

$f_{w^*}(x_i) = y_i \;\Rightarrow\; \forall_i \, L_i(w^*) = 0$

implies exponential convergence w. fixed step size



Exponential convergence of m-SGD

Convex loss function L (λ-smooth, α-strongly convex), L_i (β-smooth).

Theorem [exponential convergence of m–SGD in interpolation regime]

$$E\, L(w_{t+1}) \leq \frac{\lambda}{2}(1-\eta^*(m)\alpha)^t\, ||w_1 - w^*||$$

$$\eta^*(m) = \frac{m}{\beta + \lambda(m-1)}$$

[Ma, Bassily, **B.**, ICML 18]

Related work ($m = 1$): [Strohmer, Vershynin 09] [Moulines, Bach, 11] [Schmidt, Le Roux, 13] [Needell, Srebro, Ward 14]

Minibatch size?

$$\eta^*(m) = \frac{m}{\beta + \lambda(m-1)}$$

Theorem: Critical size $m^* = \frac{\beta}{\lambda}$ [optimal fixed step size]

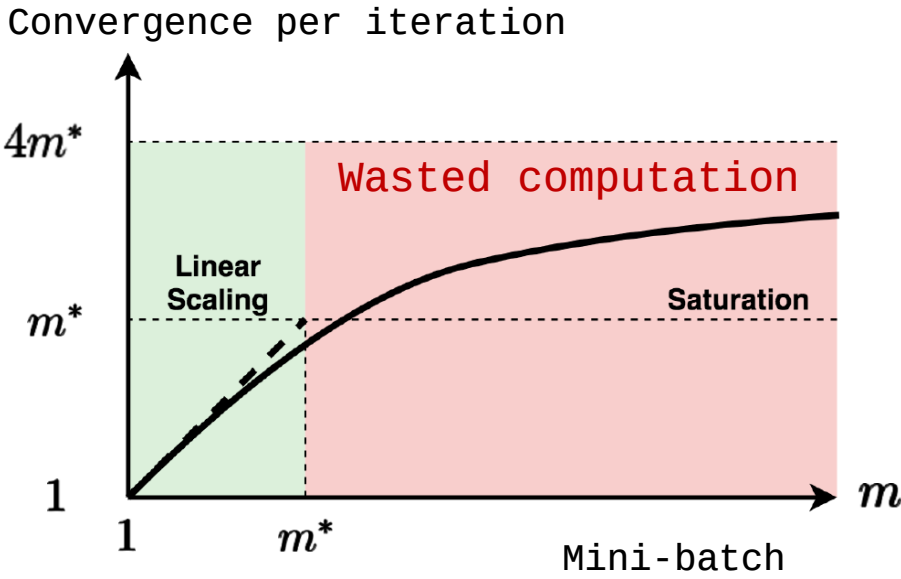
[1. linear scaling: $m \le m^*$]

One step m-SGD $\approx$ m steps of 1-SGD

(empirically observed in DNN, [Goyal, et al, 17])

[2. saturation: $m \ge m^*$]

One step m-SGD $\sim$ one step of full GD



m^* is (nearly) data independent. $O(n)$ computational gain over GD.

[Ma, Bassily, **B**., ICML 18]

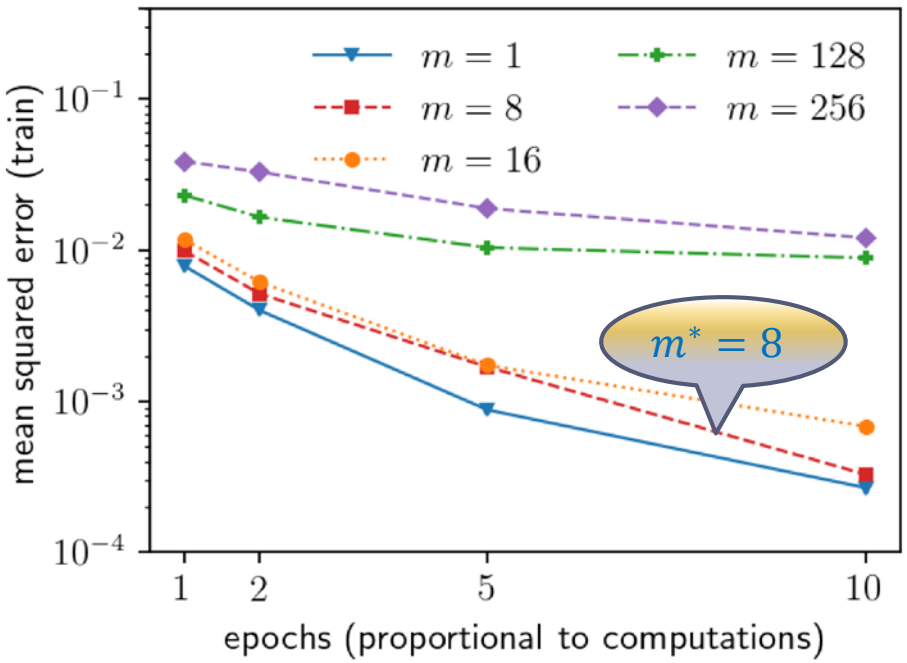
SGD is (much) faster than GD

Real data example.

One step of SGD with mini-batch $m^* \approx 8$

=

One step of GD.



[Ma, Bassily, B., ICML 18]

The power of interpolation

Optimization in modern deep learning:

- overparametrization
- interpolation
- fast SGD
- GPU → 

SGD $O\left(\frac{n}{m^*}\right)$ computational gain over GD

* GPU implementation ~ 100 over CPU.

$n = 10^6, m^* = 8$: SGD on GPU $\sim 10^7$x faster than GD on CPU!

Learning from deep learning: fast and effective kernel machines

EigenPro 2.0

Dataset	Size	Dimension	Our method (GPU)	ThunderSVM (GPU) [WSL+18]	LibSVM (CPU)
TIMIT	$1 \cdot 10^5$	440	**15 s**	480 s	1.6 h
SVHN	$7 \cdot 10^4$	1024	**13 s**	142 s	3.8 h
MNIST	$6 \cdot 10^4$	784	**6 s**	31 s	9 m
CIFAR-10	$5 \cdot 10^4$	1024	**8 s**	121 s	3.4 h

Smaller datasets take seconds.

No optimization parameters to select.

Code: https://github.com/EigenPro

[Ma, B., NIPS 17, SysML 19]

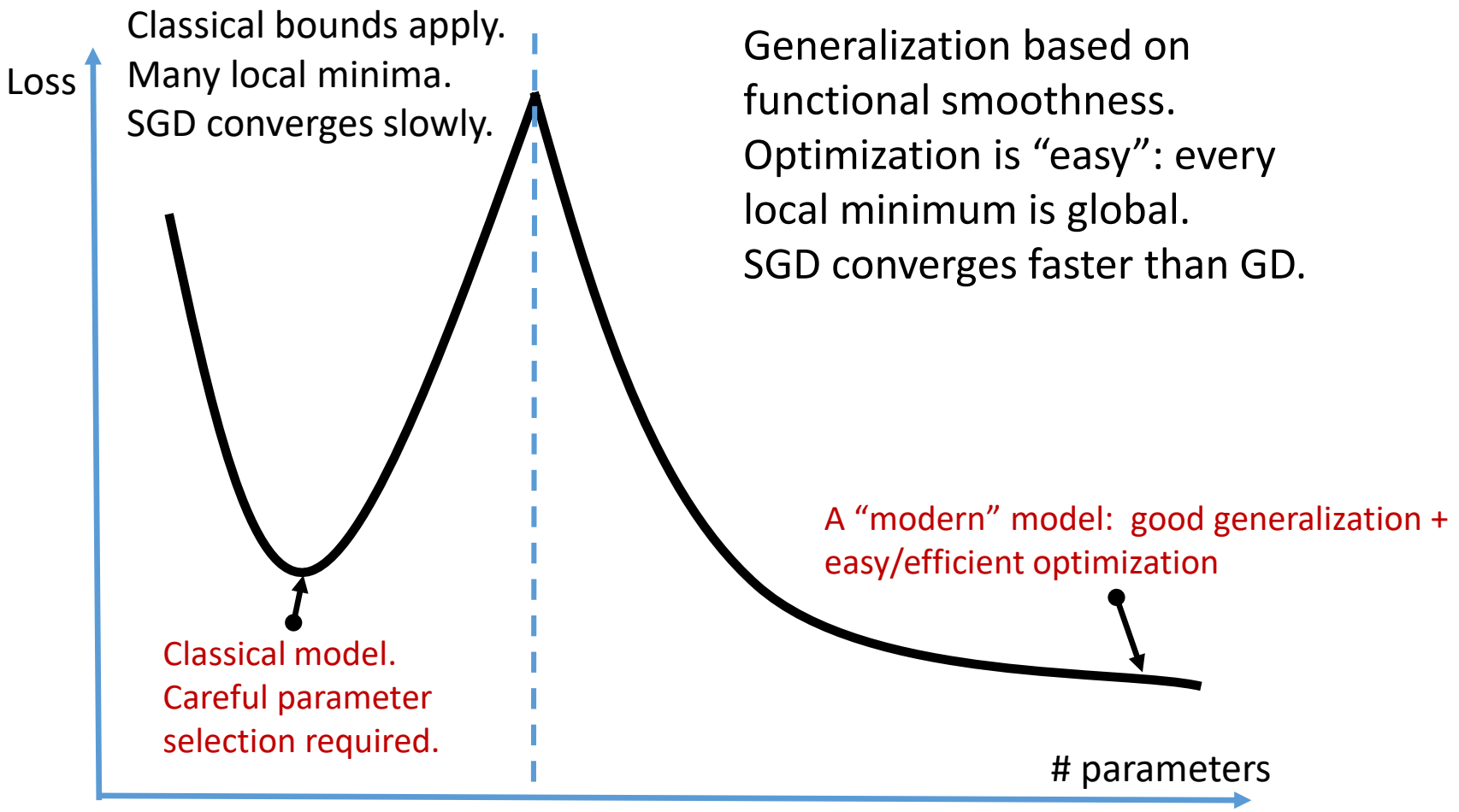
Important points

- New phenomenon is interpolation, not over-parametrization.
  - Classical methods, like kernels machines/splines are infinitely over-parametrized. Over-parametrization enables interpolation but is not sufficient.
- Empirical loss is a useful optimization target, not a meaningful statistic for the expected loss.
- Optimization is qualitatively different under interpolation.
  - Every local minimum is global.
  - SGD is overwhelmingly faster than GD.
- Many phenomena can be understood from linear regression.

From classical statistics to modern ML

Classical.

Modern ML (interpolation regime).



Collaborators:

Siyuan Ma, Ohio State University
Soumik Mandal, Ohio State University

Daniel Hsu, Columbia University
Raef Bassily, Ohio State University
Partha Mitra, Spring Harbor Labs.
Sasha Rakhlin, MIT
Sasha Tsybakov, ENSAE

Thank you